# Group Policy Introduction

## Group Policy

Group Policy allows you to setup users’ environments only once, and to rely on the operating system to enforce them thereafter.

Group Policy objects(GPO) are different from profiles. A profile is a user environment setting that a user can change desktop settings, registry settings in NTUser.dat files, profiles directory, My Documents, or Favorites. You, as the administrator, manage and maintain Group Policy, an MMC hosted administrative tool used to set policy on groups of users and computers.

## Local Group Policy

As its name implies, it is stored on a local machine at \systemroot\System32\GroupPolicy. It can be useful if you only need to apply certain settings to a small number of Windows XP or Windows 2000 clients, or your clients are not members of a domain.

## Active Directory Structure and Group Policy

Group Policy uses the Active Directory structure (referred to in this course as the Active Directory hierarchy) as a map for applying specific GPOs to specific Users and Computers. Group Policy objects are linked to Active Directory container objects…Sites, Domains, or Organizational Units (SDOUs). And any of these objects may have links to one or multiple GPOs. The location of the User or Computer account and the GPOs linked to the Site, Domain or Organizational Unit(s) in which those accounts reside determine what Policies are applied to the User or Computer.

It is important to note that GPOs are linked only from objects of objectClass=Site, objectClass=domainDNS, and objectClass=OrganizationalUnit. These objects can contain other objects. For simplicity, these objects will be referred to as Group Policy containers when used generally in this course.

## The Order of Group Policy Application

The local Group Policy object is applied first. Then site-linked Group Policy objects are applied in specified order, then domain-linked ones in specified order, and lastly organizational unit-linked Group Policy objects beginning at the highest (in Active Directory hierarchy) organizational unit containing the user or computer account and ending with the lowest (closest to the user or computer) organizational unit containing the user or computer. At each organizational unit, any Group Policy objects linked to it are applied in administratively specified order.

The order of application detailed in the previous paragraph (1. Local, 2. Site, 3. Domain, 4. Organizational Unit) is significant to the architecture of Active Directory, because by default, policy applied later overwrites policy applied earlier for each setting that is either Enabled or Disabled. Settings that are not configured do not overwrite anything — any Enabled or Disabled setting applied earlier is allowed to persist.

This is the default behavior for policy application. Mechanisms do exist that let you either force or prevent Group Policy objects from affecting groups of users or computers. The most powerful mechanisms for avoiding the default behavior are the **No Override** and **Block Policy Inheritance** settings. These settings can be configured via the GPO properties. It is best to minimize the use of these.

What happens if these two settings appear to conflict? For example: an Administrator at the OU level could set the Block Inheritance flag for that OU, which would prevent Policies from above applying to accounts in that OU or in child OUs. However, if the Domain or Enterprise Administrator has set the No Override flag on a GPO, this would trump the Block Inheritance setting at the OU level.

In order for a GPO to apply to a given user or computer, that user or computer must have both **Read** and **Apply Group Policy** permissions on the GPO. By default, Authenticated Users have both Apply Group Policy and Read permissions set to Allow. Both of these permissions are managed together as a single unit by using Security Filtering in Group Policy Management Console (GPMC).

To set the permissions for a given GPO:

1. Right click the OU and select **Properties.**
2. Select the **Group Policy** tab.
3. Select the correct Group Policy and click the **Properties** button.
4. Select the **Security** tab.

## Location of GPOs

Domain Group Policy objects store Group Policy information in two locations: a Group Policy container and a Group Policy template. They are named with a globally unique identifier (GUID), which is used to keep them synchronized.

Diagram

Description automatically generated

### Group Policy Container

The Group Policy container, located under cn=system, is an Active Directory storage area for Group Policy object properties; it includes both computer and user Group Policy information. The Group Policy container has the following properties:

* Version Information

This makes sure that the information is synchronized with the Group Policy template information.

* Status Information

This indicates whether the Group Policy object is enabled or disabled.

* Components with Settings in GPO

List of components (extensions) that have settings in the Group Policy object.

* Policy Settings as Defined by the Extension Snap-ins

This data repository contains data for all applications, interfaces, and APIs that provide for application publishing and assigning.

### Group Policy Template

Group Policy template is located in the System Volume folder of domain controllers (Sysvol) in the \Policies subfolder. The Group Policy template is the container where Administrative Template–based policy settings, Security Settings, applications available for Software Installation, and script files are stored.

When you modify a Group Policy object, the directory name given to the Group Policy template is the GUID of the Group Policy object that you modify.

For example, a Group Policy template folder might be named as shown in the following example:

%systemroot%\sysvol\SYSVOL\www.Reskit.com\Policies\{47636445-af79-11d0-91fe-080036644603}

A Group Policy snap-in can store data outside the Group Policy object; however, this requires that at least a link to the Group Policy object be stored either in a Group Policy container (Active Directory data store) or in a Group Policy template (file-type data stored on the Sysvol folder).

### Viewing and Reporting of Policy Settings

When multiple GPOs apply to a given user or computer, they can contain conflicting policy settings. For most policy settings, the final value of the policy setting is set only by the highest precedent GPO that contains that setting. Resultant Set of Policy (RSoP) helps you understand and identify the final set of policy that is applied as well as settings that did not apply as a result of policy inheritance.

Specifically, Resultant Set of Policy helps you determine:

* The final value of the setting that is applied as a result of all the GPOs.
* The final GPO that set the value of this setting (also known as the winning GPO).
* Precedence details that show any other GPOs that attempted to set this setting and the value that each GPO attempted to set for that policy setting.

Group Policy Management Console addresses some common reporting requirements including the ability to document all the settings in a GPO to a file for printing or viewing.

# Delegating Administration of Group Policy

Organizations need to be able to delegate administration of Group Policy to other administrators who can take responsibility for a given OU, domain, or other container. Active Directory is designed to allow you to delegate control of portions of the directory service in managing aspects of Group Policy. The following areas can be delegated:

* GPO delegation. This includes permission to **create** GPOs in a domain or permission to **edit** an existing GPO. Note that having permission to edit a GPO **does not** include any delegated rights on the GPO links.
* Link delegation. This includes permission to add, delete, or change links to GPOs. Note that having link delegation **does not** include any delegated rights on the GPO itself.
* RSOP delegation. This includes permission to run RSoP (in either planning or logging mode) on objects under a container.
* WMI filter delegation. This includes permission to create WMI filters or permission to edit an existing filter.

In GPMC, delegation is simplified because it manages the various Access Control Entries (ACEs) required for a task as a single bundle of permissions for the task. You can also use the Access Control List (ACL) editor to view or manage these permissions manually.

The underlying mechanism for achieving delegation is the application of the appropriate DACLs to GPOs and other objects in Active Directory. This mechanism is identical to using security groups to filter the application of GPOs to various users. You can also specify Group Policy to control who can use MMC snap-ins. For example, you can use Group Policy to manage the rights to create, configure, and use MMC consoles, and to control access to individual snap-ins.

# What Is User Group Policy Loopback Mode?

Group Policy applies to the user or computer in a manner that depends on where both the user and the computer objects are located in Active Directory. In some cases, this processing order may not be appropriate (for example, when you do not want applications that have been assigned or published to the users in their OU to be installed while they are logged on to the computers in some specific OU). With the Group Policy loopback support feature, you can specify two other ways to retrieve the list of GPOs for any user of the computers in this specific OU:

* **Merge Mode**

In this mode, when the user logs on, the user's list of GPOs is gathered normally. Then the computer’s list of GPOs is gathered using the computer's location in Active Directory. The list of GPOs for the computer is then added to the end of the GPOs for the user. This causes the computer's GPOs to have higher precedence than the user's GPOs.

* **Replace Mode**

In this mode, the user's list of GPOs is not gathered. Only the list of GPOs based on the computer object is used.

# Group Policy Dependencies

Group Policy has several key dependencies.

Domain-based Group Policy requires an Active Directory environment with DNS properly configured.

## Active Directory

In a network environment, Group Policy depends on Active Directory as the targeting framework that allows you to link GPOs to specific Active Directory containers such as sites, domains, or OUs.

In a stand-alone environment without Active Directory, you can use Local Group Policy objects to configure settings on individual computers.

## Domain Name System (DNS)

**Group Policy application** requires clients to access specified servers, including domain controllers and other servers such as share points and install points.

**Group Policy management** also requires access to domain controllers. If the network is functioning, but clients or consoles such as the Group Policy Object Editor or GPMC are unable to locate the servers, there might be a problem with your network's DNS system.

## Replication

Group Policy depends on other technologies in order to properly replicate between domain controllers in a network environment. A GPO is a virtual object stored in both Active Directory and the Sysvol of a domain controller.

**Property settings**, stored in the Group Policy container, are replicated through Active Directory replication. Replication automatically copies the changes that originate on a writable directory partition replica to all other domain controllers that hold the same directory partition replica. More specifically, a destination domain controller pulls these changes from the source domain controller.

**Data settings**, stored in the Sysvol as the Group Policy template, are replicated through the File Replication Service (FRS), which provides multi-master file replication for designated directory trees between designated servers running Windows Server 2003.

**The Group Policy container** stores GPO properties, including information about version, GPO status, and a list of components that have settings in the GPO.

**The Group Policy template** is a directory structure within the file system that stores Administrative Template-based policy settings, security settings, script files, and information regarding applications that are available for software installation. The Group Policy template is located in Sysvol in the \Policies sub-directory for its domain.

GPOs are identified by their globally unique identifiers (GUIDs) and stored at the domain level. The settings from a GPO are only applied when the Group Policy container and Group Policy template are synchronized.

## DFS publishing

The Sysvol folder is shared on each domain controller and is accessible through the UNC path \\dcname.domainname\sysvol.

The Sysvol is also published as a domain-based Distributed File System (DFS) share. This allows clients to access the Sysvol by using the generic path \\domainname\sysvol. A request for a DFS referral for \\domainname\sysvol will always return a replica in the same Active Directory site as the client if one is available. This is the mechanism that the Group Policy client-side extensions use to retrieve a local copy of the Group Policy template information.

# Core Group Policy Physical Structure

Understanding where GPOs are stored and how they are structured can help you troubleshoot problems you might encounter when you implement Group Policy. Although GPOs can be linked to sites, domains, and OUs, they are stored only in the domain. As explained earlier, a GPO is a virtual object that stores its data in two locations: a Group Policy container and a Group Policy template.

## Group Policy Container

A Group Policy container is a location in Active Directory that stores GPOs and their properties. The properties of a GPO include both computer and user Group Policy information. The Policies container is the default location of GPOs. The path to the Policies container, in Lightweight Directory Access Protocol (LDAP) syntax, is CN=Policies,CN=System,DC=Domain\_Name, where the Domain\_Name values specify a fully qualified domain name (FQDN).

Some of the information in a Group Policy container includes:

* Version information

Ensures that the information is synchronized with the Group Policy template information.

* Status information

Indicates whether the user or computer portion of the GPO is enabled or disabled.

* List of components

Lists (extensions) that have settings in the GPO. These attributes are gPCMachineExtensionNames and gPCUserExtensionNames.

* **File system path**

Specifies the Universal Naming Convention (UNC) path to the Sysvol folder. This attribute is gPCFileSysPath.

* **Functionality version**

Gives the version of the tool that created the GPO. Currently, this is version 2. This attribute is gPCFunctionalityVersion.

* **WMI filter**

Contains the distinguished name of the WMI filter. This attribute is gPCWQLFilter.

## System Container

Each Windows Server 2003 domain contains a System container. The System container stores per-domain configuration settings, including GPO property settings, Group Policy container settings, IP Security settings, and WMI policy settings. IP Security and WMI policy are deployed to client computers through the GPO infrastructure.

The following subcontainers of the System container hold GPO-related settings:

* Policies.   
  This object contains groupPolicyContainer objects listed by their unique name. Each groupPolicyContainer object holds subcontainers for selected computer and user policy settings.
* Domain, OUs and Sites.   
  These objects contain two GPO property settings, gPLink and gPOptions.
* Default Domain Policy.   
  This object contains the AppCategories container, which is part of the Group Policy Software installation extension.
* IP Security.   
  This object contains IP Security policy settings that are linked to a GPO. The linked IP Security policy is applied to the recipients (user or computer) of the GPO.
* WMIPolicy.   
  This object contains WMI filters that can be applied to GPOs. WMI filters contain one or more Windows Query Language (WQL) statements.

## System\Policies Container

The System container is a top level container found in each domain naming context. It is normally hidden from view in the Active Directory Users and Computers snap-in but can be made visible by selecting “Advanced Features” from the snap-in View menu inside MMC. (showInAdvancedViewOnly = TRUE.) Group Policy information is stored in the Policies subcontainer of this container. Each GPO is identified by a GroupPolicyContainer object stored within the Policies container.

The Group Policy container is located in **the Domain\_Name/System/Policies container**. Each Group Policy container is given a common name (CN) and this name is also assigned as the container name.

The default GPOs are assigned the same Group Policy container CN on all domains. All other GPOs are assigned a unique CN. The default GPOs and their Group Policy container common names are:

• **Default Domain Policy**: {31B2F340-016D-11D2-945F-00C04FB984F9}.

• **Default Domain Controllers Policy**: {6AC1786C-016F-11D2-945F-00C04fB984F9}.

Knowing the common names of the default GPOs will help you distinguish them from non-default GPOs.

# How a Group Policy Container is Named

Group Policy containers are named automatically when they are created. The CN of each Group Policy container is a GUID (Globally Unique Identifier). This is distinct from and unrelated to the Object GUID given to each Active Directory object. The CN is the name of the Group Policy container used to ensure uniqueness of Group Policy container names within the Policies container. There is no requirement for these GUIDs to be unique between domains (the Default Domain Policy and the Default Domain Controllers Policy GPOs each have identical GUIDs in all Active Directory installations). However, an Object GUID is always unique across all installations of the Active Directory store.

The following table shows permissions on Group Policy container:

**Default Group Policy Container Permissions**

|  |  |
| --- | --- |
| **Trustee** | **Access** |
| Authenticated Users | Read, Apply Group Policy |
| Domain Admins | Read, Write |
| System | Read, Write |

## GPO Attributes in the Policies CN

GPOs are created by instantiating the groupPolicyContainer class in the Active Directory schema and storing the resulting GPO in the System/Policies container of the Active Directory store. After creating a GPO, you can review its CN from the Active Directory Users and Computers snap-in. You can review all GPO attributes and their values from the ADSI Edit.

**Mandatory Attributes of the groupPolicyContainer Class**

|  |  |
| --- | --- |
| **Name** | **Description** |
| CN | The common name of the GPO. This is in the form of a GUID to avoid GPO naming conflicts within the Policies container. |
| instanceType | An attribute that dictates how an object is instantiated from its class on a particular server. In this case, it describes how the groupPolicyContainer class is created into a GPO in the Active Directory. A GPO is assigned the instanceType value of 4. |
| objectCategory | An object class name, including the object's path, used to group object's of the instantiated class. For example, the objectCategory of a GPO in the contoso.com domain is: **CN=Group-Policy-Container, CN=Schema, CN=Configuration, DC=contoso,DC=com**. |
| objectClass | The list of classes from which this class is derived. For a GPO, the objectClass is **Container, groupPolicyContainer, and top**. |

There are also a number of optional attributes **inherited** from the top class, and others that are assigned directly to the Group Policy container. Many optional attributes are required in order for the Group Policy container to function properly. For example, the **GPCFileSysPath** optional attribute must be present or the Group Policy container will not be linked to its corresponding Group Policy template.

## GroupPolicyContainer Subcontainers

Within the GroupPolicyContainer there are a series of subcontainers. The first level of subcontainers — **User** and **Machine** — belong to the class Container. These two containers are used to separate some User-specific and Computer-specific Group Policy components.

## Group Policy Container-Related Attributes of Domain, Site, and OU Containers

Windows Server 2003 uses domain, DNS, site, and organizational unit classes to create domain, site, and OU container objects respectively. These objects contain two optional Group Policy container-related attributes, **gPLink** and **gPOptions**.

* **gPLink** property contains the prioritized list of GPOs.   
  The **gPLink** attribute holds a list of all Group Policy containers linked to the container and a number for each listed Group Policy container, that represents the **Enforced** (previously known as No Override) and **Disabled** option settings. The list appears in priority order from lowest to highest priority GPO.
* **gPOptions** property contains the Block Policy Inheritance setting.   
  The **gPOptions** attribute holds an integer value that indicates whether the Block Policy Inheritance option of a domain or OU is enabled (1) or disabled (0).

## Managing Group Policy Links for a Site, Domain, or OU

To manage GPO links to a site, domain, or OU, you must have read and write access to the **gPLink** and **gPOptions** properties. By default,

* **Domain Admins** have this permission for domains and organizational unit
* Only **Enterprise Admins** and **Domain Admins of the forest root domain** can manage links to sites.

Active Directory supports security settings on a per-property basis. This means that a non-administrator can be delegated read and write access to specific properties.

* E.g., if non-administrators have read and write access to the gPLink and gPOptions properties, they can manage the list of GPOs linked to that site, domain, or OU.

## How WMIPolicy Objects are Stored and Associated with Group Policy Container Objects

A single WMI filter can be assigned to a Group Policy container. The Group Policy container stores the distinguished name of the filter in **gPCWQLFilter** attribute. The Group Policy container locates the assigned filter in the **System/WMIPolicy/SOM** container.

Each Windows Server 2003 domain stores its WMI filters in this Active Directory container. Each WMI filter stored in the SOM container lists the rules that define the WMI filter.

Each rule is listed separately. For example, consider a WMI filter containing the following three WQL queries:

SELECT \* FROM Win32\_Product WHERE IdentifyingNumber = "{5E076CF2-EFED-43A2-A623-13E0D62EC7E0}"

SELECT \* FROM Win32\_Product WHERE IdentifyingNumber = "{242365CD-80F2-11D2-989A-00C04F7978A9}"

SELECT \* FROM Win32\_Product WHERE IdentifyingNumber = "{00000409-78E1-11D2-B60F-006097C998E7}"

Three WMI rules are defined in the details of the filter. Each rule contains a number of attributes, including the query language (WQL) and the WMI namespace queried by the rule.